AMENDMENTS TO THE SPECIFICATION:

Please replace the paragraph beginning at line 22 on page 1 with the following amended paragraph.

This invention advantageously uses Applicants' Trusted Custodial Utility that holds electronic original records and holds comparable system roles as a virtual electronic vault in validating the right of an individual to perform a requisite action, the authenticity of submitted electronic information objects, and the status of the authentication certificates used in the digital signature verification and user authentication processes. Such TCUs and operations are described in U.S. Pat. No. 5,615,268; No. 5,748,738; No. 6,237,096; and No. 6,367,013.

Please replace the paragraph beginning at line 19 on page 2 with the following amended paragraph.

[[The]] Legal standing for electronic signatures applied to information objects is made possible by the passage of U.S. Electronic Signatures in Global and National Commerce Act (ESIGN) legislation and U.S. state laws modeled after the UETA drafted by the National Conference of Commissioners on Uniform State Laws and approved and recommended for enactment in 1999 ~~provide assurances of legal standing for electronically signed information objects (electronic documents) which has generated~~ that has resulted in government, banking and electronic commerce ~~activity~~ activities aimed at realizing the efficiency and economies of these potentially wholly electronic transactions.

Please replace the paragraph beginning at line 32 on page 12 with the following amended paragraph.

The CSS is provided with and maintains sufficient information on the location, the means of communication, and the means of processing certificate status for every CA that it needs to interoperate with. The CSS therefore makes it possible to stabilize and optimize the application design. The CSS advantageously parses and caches certificate status to minimize status response time to a TCU status query. The CSS therefore eliminates the need for any of the traditional forms of PKI interoperability. Potential compromise recovery is greatly enhanced since a TCU user account can easily be deactivated or a set of users eliminated by removing the CA from the CSS list of approved CAs.

Please replace the paragraph beginning at line 20 on page 14 with the following amended paragraph.

Whenever a digital signature is applied, the signer will be requested to affirm their intent to be bound by their digital signature. This commit action, that is required by recent legislation, may take the form of readable text in a display window or splash screen, and may require invocation of a graphical button and/or logon to a cryptographic token that is also a cryptographic key and certificate store. The actual demonstration of said willingness to commit is through the use of a trusted application that computes the user's digital signature using the selected content and combines it with their authentication certificate to [[the]] form a signature block. The signature block may also

contain authenticated and unauthenticated data elements. Any authenticated ~~Authenticated~~ data elements that are included in the digital signature computation, such as rationale for signing or local date-time, are ~~(e.g., local date-time) and may be considered~~ protected by the digital signature ~~(integrity)~~. Unauthenticated data elements are added after the signature computation and are not protected. FIG. 4 shows a sample syntax that contains the data elements and layout of a signature block. It is not to be interpreted literally as it is only meant to be an illustrative example.

Please replace the paragraph beginning at line 1 on page 15 with the following amended paragraph.

The information object and any signature blocks may be advantageously placed in a wrapper (S/MIME) or at tags in an extensible information syntax (XML, HTML, XHTML) for handling convenience and to facilitate information processing. This data structure is then sent to the TCU for validation. ~~Conversely~~ Alternatively, the signature block(s) may be sent independently to the TCU to be affixed to the actual source record which never leaves the TCU. In the latter case, each signature block is validated separately.

Please replace the paragraph bridging page 16 and 17 with the following amended paragraph.

In the highly assured environment in which the TCU is operated, certificate status checking is only needed when a service is requested by a

qualified user. For information objects, certificate status need only be checked at the time of submission. If all digital signatures are determined to be valid, the information object is deemed authentic thereafter. Security and procedural practices and methods are in place at the TCU to prevent malicious actions and hardware failures that result in unauthorized document alteration or loss. Every submission results in creation of a new version of an electronic source record. The TCU is charged with maintaining knowledge as to which is the latest version of the source record. This version may be identified as the electronic original and as a transferable record. The TCU demonstrates its assumption of control of an original source record by adding a reliable date-time stamp to the source record and then by applying its digital signature and appending its certificate. A wrapper may be applied to the source record for security and processing expediency. Although this versioning process creates a standalone authenticated trail-of-evidence and custody, separate redundant audit records are maintained for corroboration.

Please replace the paragraph bridging page 17 and 18 with the following amended paragraph.

Where CRLs are used, the CSS retrieves the latest rendition of the CRL from the CA distribution point, e.g., an X.509 v2 CRL profile (IETF RFC2459, Jan 99), validates its signature, parses it, and creates a cache to store the results. The CSS uses a CA's CRL publication interval to govern when it performs the next CRL download. Every CRL contains a validity field that is normally set to allow some leeway in performing downloads. This

allows for communications congestion and CA downtime and will force the CSS to require remedial action if this interval is exceeded. Such remedy may include revalidating any submissions that are associated with a newly added revoked certificate. Each new CRL supersedes the previously loaded CRL. The exception to this rule is for delta CRLs procession. The contents of a delta CRL are appended to the current cache contents. The delta CRL Base CRL Number refers to the most recent full CRL issued. Delta CRLs are published at shorter intervals (minute, hour) and only when a certificate revocation has occurred since the last full CRL. The CSS is responsible for retrieving CRLs and delta CRL based on publication interval or notification ~~and not to exceed the interval established in the TCU security policy~~.

Please replace the paragraph beginning at line 9 on page 18 with the following amended paragraph.

A second method used by CAs to distribute certificate status is the OCSP. Where OCSP is used the CSS queries the OCSP responder when asked for certificate status. OCSP responses are signed to guarantee their integrity and authenticity. The CSS parses the OCSP response and adds certificate details and status to another cache. A time-to-live flag, determined by local TCU security policy, is included with and determines when the entry will be removed from the cache. This feature is aimed at minimizing communications overhead when several information objects are [[be]] uploaded by the same party/entity to the TCU in a short interval. The time-to-live flag will usually be significantly shorter (e.g., 5 minutes) than the normal

CRL publishing interval (twice daily, daily). The CSS may check certificate status again, if more than one information object was processed, prior to purging certificate status from the cache to ensure that certificate revocation has not occurred. If certificate revocation has occurred during the time-to-live interval, then the owner organization point of contact must be notified. Several other query methods exist, but will not be described for brevity. Be it understood that they will each require a connector and potentially a separate cache when they are utilized.

Please replace the paragraph bridging page 18 and 19 with the following amended paragraph.

Having previously created, executed or retrieved the electronic document, a submitter digitally signs and submits it to the TCU as in step 101. In this eSeal process, a wrapper that contains the signed content and digital signature block(s) that further contain the digital signature(s) and certificates(s) of the submitter and any other signatory is formed. There are five processes represented in FIG. 1: (1) action when an invalid digital signature(s) and/or revoked certificate(s) is found, (2) certificate status checking where status is locally cached, (3) certificate status checking where certificate status has to be retrieved, (4) CRL retrieval and processing, and (5) creating an eOriginal when the eSeal document is determined to be authentic. In step 103 the TCU receives the eSealed electronic document. In step 105 the TCU validates that the submitter has authority to add the electronic document to a selected account and/or transaction. In step 107, the TCU

cryptographically verifies any digital signatures included in the electronic wrapped digital electronic document. The public key, found in the signer's X.509 authentication certificate, is used during the verification process. In step 109, the certificate validity period is extracted from the signer's authentication certificate, and in step 111, the validity period is checked against the current date and time. If any of the before mentioned tests fail, the submission is rejected in step 113 and a negative acknowledgment may be sent in step [[114]] 115. The action is logged in step 117.

Please replace the paragraph beginning at line 27 on page 20 with the following amended paragraph.

CRLs are published in [[step]] steps 155 and 159 at predetermined intervals and in step 157 as needed when a suspected compromise is reported and policy requires an immediate response. This process is further described in FIG. 2.

Please replace the paragraph beginning at line 11 on page 21 with the following amended paragraph.

In [[step]] steps 155 and 159, a CA Administrator configures the CA to publish CRLs at predetermined intervals. In step 157, the CA Administrator may also publish a Delta CRL as dictated by the local certificate or security policy. The CA Administrator or CA will push notice on publication of a Delta CRL. A Delta CRL may be generated whenever a certificate is revoked or suspended during the interval between publications of the full CRLs. Delta

CRLs may contain a complete list of revoked CRLs. In step 201, CRLs and Delta CRLs are published to a CRL repository or directory.

Please replace the paragraph beginning at line 18 on page 21 with the following amended paragraph.

In step 203, the CSS retrieves the CRL publication schedule or Delta CRL notice, and step 205 represents a timer used for scheduled retrieval. The timer also allows retrieval based on the "next update" field contained in all CRLs. In step 207, the CRL or Delta CRL is retrieved from the CRL repository. In step 209, the CRL or Delta CRL is parsed prior to being added in step 153 to an appropriate cache or list in the Certificate Status Store in step 121 ~~or directory~~ based on the established schedule or upon notification. Parsing the CRLs allows for easier management and reduced overhead in CRL entry lookup. Steps 119, 123, 125, 135, 137, and 141 of the CSS are illustrated in FIG. 2 for completeness, and are implemented as described in connection with FIG. 1.

Please replace the paragraph beginning at line 3 on page 27 with the following amended paragraph.

It will be recognized that the Application Server and associated Electronic Vault may be used by the dealer to stage the contract for remote signing by the lessor. In steps 807, 809, and ~~through~~ 811, the lessor retrieves the lease from the vault, agrees to the terms of the lease by digitally signing it, and returns its digitally signed version to the vault. Steps 807, 809, and

~~through~~ 811 illustrate both multi-site collaboration and asynchronous transaction processing.

Please replace the paragraph beginning at line 9 on page 27 with the following amended paragraph.

In steps 813, 815 and ~~through~~ 817, the received electronic document(s) (the lease) are checked for digital signatures, and if any are found, the digital signatures are verified and the respective authentication certificates are validated. In step 817, the local time is checked to ensure that it falls within the validity period(s) of the certificate(s), and in step 819, the CSS is queried for the status of the certificate(s). In response in step 821, the CSS first checks its local cache memory or data store for certificate status, and if a certificate's status is present and current, the CSS returns the certificate's status as "active" in step 827. In step 823, if certificate status is not present or not current, the CSS queries the issuing CA using the connector type created for this purpose. In step 825, the issuing CA, e.g., the Bank CA, or its status reporting means (e.g., directory) returns status to the CSS, preferably using the same connector, and in step 827, the CSS reports the queried certificate's status back to the Application Server.

Please replace the paragraph beginning at line 21 on page 27 with the following amended paragraph.

Assuming all digital signatures and certificates are verified and validated, proving the electronic document authentic, the Application Server

assumes control of the electronic document and saves it in the Electronic Vault as a new version in step 829. Thus, it will be seen that, with the proper characteristics, the Application Server and Electronic Vault cooperate as a TCU. In step 831, the new version is designated as an authoritative copy, an ~~Electronic Original~~ electronic original record that may also be a transferable record, by appending a date-time stamp and applying the TCU's digital signature to the document. As long as at least one digital signature on a document is valid, this step takes place.

Please replace the paragraph beginning at line 4 on page 29 with the following amended paragraph.

The transaction execution method further includes the steps of requiring any signing entity to commit to use of and to be bound by their digital signature prior to the act of signing, executing said information object by any party~~, consists of inclusion of~~ by applying at least the digital signature and authentication certificate of the signing party, creating a signature block that contains at least the digital signature and authentication certificate of the signing party, associating the signature block with the information object, repeating the previous execution steps where multiple entities digitally sign the information object and/or wrapper, and forwarding the digitally signed and/or wrapped information object to a TCU. The TCU verifies every digital signature and validates each associated authentication certificate and retrieves status from a CSS. The signature blocks are rejected if the signer's digital signature does not verify or a signer's authentication certificate has

expired or is reported to be revoked. Rejection of any signature block results in a request for a replacement signature block or initiation of remedy. If at least one signature block is determined to be valid, the TCU appends its own signature block, also containing reliable date and time, to the subject information object, creating an electronic original which it holds and controls on behalf of the owner.

Please replace the paragraph beginning at line 20 on page 29 with the following amended paragraph.

Creating a digital signature block may include the steps of computing one or more content hashes for the one or more information object fragments or for the whole information object, computing a hash over the one or more content hashes and any appended data, such as the local date and time, signing rationale, or an instruction, encrypting the computed hash using the signing party's private key, thereby forming the signer's digital signature, and placing the signer's digital signature in the signature block along with the signer's authentication certificate. If the appended data includes a local date and time, creating a digital signature block may further include the steps of either displaying the local date and time, requiring a signer to affirm that the date and time are correct, and correcting the local date and time if either is incorrect, or relying on a system date and time if these are set by a trusted time service and local date and time [[is]] are protected from tampering. The local date and time can be checked to ensure that [[it]] they [[is]] are accurate and that ~~it falls~~ they fall within the user's authentication certificate validity

period and that the local data and time [[is]] are not before and not after the dates and times specified by the validity period.

Please replace the paragraph beginning at line 3 on page 30 with the following amended paragraph.

Remedy of a digital signature that fails to verify requires the digital signature to be recomputed and the signature block to be retransmitted. Remedying a violation of the authentication certificate validity period includes notifying the user that the user's certificate has expired and must be renewed and notifying the transaction owner that the transaction is incomplete.

Please replace the paragraph beginning at line 8 on page 30 with the following amended paragraph.

Placement of one or more signature blocks and the information contained therein is specified by at least one signature tag. One or more handwritten signatures and dates are digitized and used for information object execution, and placement of the signatures and dates is specified by at least one signature tag. One or more signature blocks can be sent to the TCU separately along with the designation of the corresponding signature tags and the TCU can validate every signature block sent independently or as a group. If either the digital signature verification or authentication certificate validation step fails, the TCU rejects the signature block and may request remedy, and if the signature block validation step succeeds, the TCU places the signature block at the indicated tag. To signature blocks sent separately, the TCU may

add a reliable date and time to each signature block. According to business rules, the TCU appends its own signature block that contains a reliable date and time in a wrapper that encompasses the subject information object and inserted signature block fields, thereby creating an electronic original information object. Multiple user ~~signatures~~ signature blocks may be added within a wrapper, and wrappers can be recursively applied to implement other business and security processes.

Please replace the paragraph bridging page 30 and 31 with the following amended paragraph.

The TCU may validate the digital signature(s) and authentication certificate(s) present in a signature block(s) that is/are to be contained within or is/are to be added to content of an electronic original information object by checking in the business rules database that the signing entity identified by the authentication certificate has authority to perform the requested action, verifying the signing entity's digital signature, checking that certificate validity period overlaps current reliable date and time, checking that the conveyed local date and time ~~falls~~ fall within allowable deviation with the TCU date and time, and checking certificate status using a CSS. If any of these steps results in an invalid or false output, the digital signature is deemed invalid, the requested action is disallowed and remedy sought; otherwise, the digital signature is deemed valid and the requested action is allowed.

Please replace the paragraph beginning at line 13 on page 31 with the following amended paragraph.

Certificate status checking advantageously employs a CSS for establishing communications, retrieving and caching certificate status from approved certificate issuing CAs. When the CSS receives a certificate status query from a system or TCU, the CSS first checks its local cache to see if the certificate status is present and if found and within the time-to-live interval, returns the status. If the certificate status is not present or is outside the time-to-live interval, then the CSS retrieves the status by first requesting the connection information from its configuration store. The CSS then establishes a communications session with the certificate status reporting component identified in its configuration store. The CSS composes a certificate status request as per the method contained in the CSS configuration store, and the CSS retrieves certificate status from the certificate status reporting component and closes the session with the component. The CSS adds at least the certificate's ID, certificate status and time-to-live to its cache and returns certificate status to the requesting system or TCU.

Please replace the paragraph bridging page 32 and 33 with the following amended paragraph.

A method for enrolling users in a system or TCU where ~~certificate~~ certificates are issued by an approved issuing CA that is known to a CSS includes vetting the user using established membership procedures and criteria, entering user enrollment information that has also been signed by an

approved organization sponsor, and creating and sending a certificate request to the identified issuing CA. The user's authentication certificate is retrieved, issued, and placed on a token for delivery. Digital signature, digital signature verification and the CSS certificate status check are performed to ensure that public-key pair generation and certificate issuance process were completed correctly. The user is required to sign the user acceptance agreement that commits the user to give the same weight to use of their digital signature as they give to use of his or its hand written signature, the token is delivered to the user, and the user's system or TCU account is activated.